

4. Präsenzübung „Algorithmen und Datenstrukturen“

Sommersemester 2009

Kartenspiel: Laden Sie sich die Vorlage für „Spielkarten“ von der Webseite zur Vorlesung herunter und schneiden Sie die 18 Karten aus.

Wiederholen Sie anschließend folgende Schritte für jedes der Sortierverfahren „*InsertionSort*“, „*SelectionSort*“, „*BubbleSort*“ und „*MergeSort*“:

- i) Notieren Sie sich die Laufzeit des Sortierverfahrens in O-Notation.
- ii) Mischen Sie die Spielkarten.
- iii) Sortieren Sie die Spielkarten. Zählen Sie dabei die Anzahl von „Schritten“.
- iv) Vergleichen Sie diese Schrittzahl mit der vorher in O-Notation angegebenen oberen Schranke.

SelectionSort-Variante: Modifizieren Sie den SelectionSort-Algorithmus so, dass das sortierte Feld vom Anfang her aufgebaut wird. Schreiben Sie ihren modifizierten Algorithmus in Pseudocode auf.

ShakerSort: *ShakerSort* ist eine Abwandlung des in der Vorlesung behandelten Sortieralgorithmus *BubbleSort*:

```
static <E extends Comparable<E>> void shakerSort(E[] data) {
    boolean swapped;
    int links = 1;
    int right = data.length-1;
    int done = right;
    do {
        swapped = false;
        for (int down = right; down >= links; down--) {
            if (data[down].compareTo(data[down-1]) < 0) {
                swapped = true;
                done = down;
                final E temp = data[down-1];
                data[down-1] = data[down];
                data[down] = temp;
            }
        }
        links = done + 1;
        for (int up = links; up <= right; up++) {
```

```

        if (data[up].compareTo(data[up-1]) < 0) {
            swapped = true;
            done = up;
            final E temp = data[up-1];
            data[up-1] = data[up];
            data[up] = temp;
        }
    }
    right = done - 1;
} while (swapped);
}

```

- Beschreiben Sie die Vorgehensweise des Sortieralgorithmus und stellen Sie die Unterschiede zu Bubblesort dar.
- Schätzen Sie grob die worst-case Laufzeit des Algorithmus ab (keine genaue Analyse).
- Ist ShakerSort stabil?
- Ein (Sortier-)Algorithmus arbeitet in-place, wenn er zusätzlich zu dem Speicherplatz, den die zu bearbeitenden Daten einnehmen, nur eine konstante Menge an Speicher (also unabhängig von der zu bearbeitenden Datenmenge) benötigt. Arbeitet ShakerSort in-place?
- In der gegebenen Java-Implementierung wird das Interface Comparable verwendet. Welche Methoden stellt dieses Interface zur Verfügung? Welchen Vorteil hat diese Implementierung im Vergleich zu einer Implementierung z.B. mit Integers als zu sortierende Objekte?

MergeSort-Variante: Man kann den Mergesort-Algorithmus so variieren, dass das Array in jeweils vier (statt zwei) etwa gleich große Teilfelder aufgespalten wird. Zum Mischen bestimmt man wiederholt das Minimum der vier vorderen Elemente und fügt dieses in das Ergebnisfeld ein. Geben Sie für die Laufzeit des Algorithmus eine Rekursionsformel an und leiten Sie daraus eine explizite Formel in O-Notation her. Es darf vorausgesetzt werden, dass $n = 4^k$ gilt.